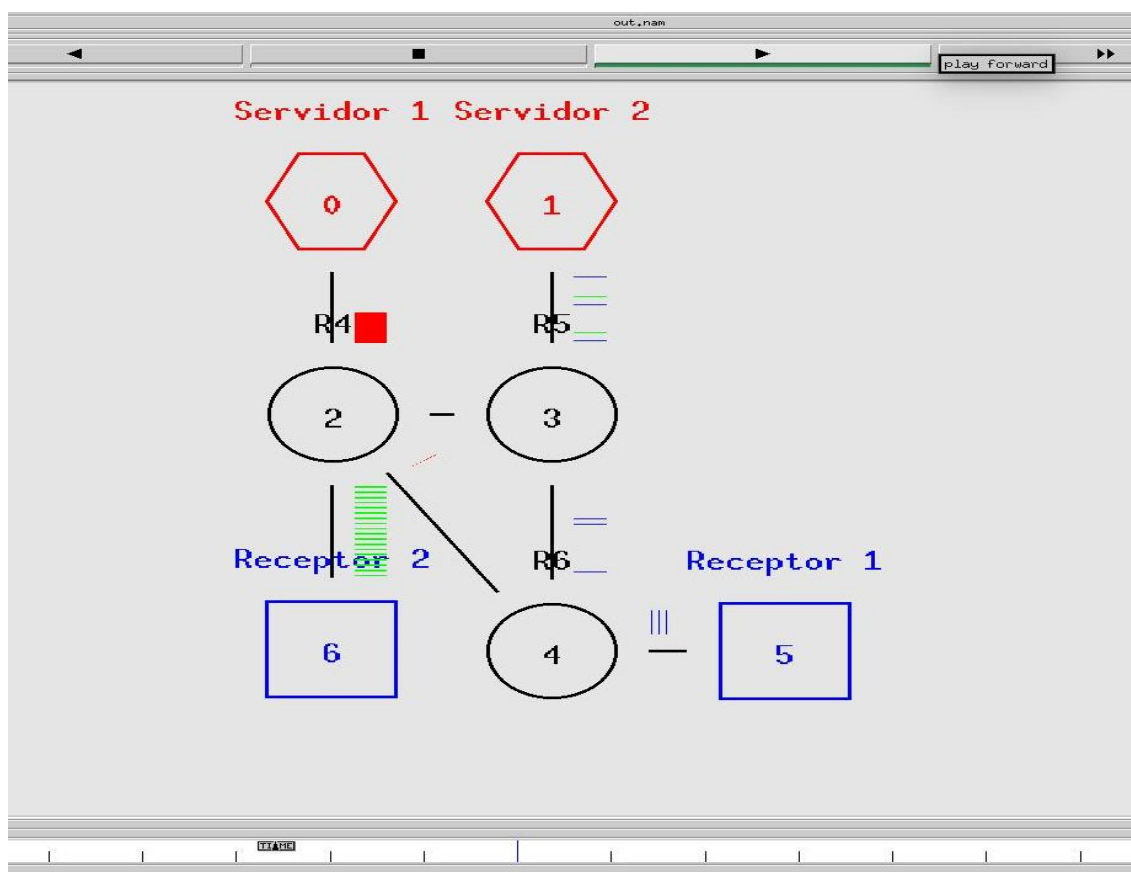


Introdução às Redes de Comunicação

Trabalho 1 –

Protocolos de Camada de Transporte



Grupo de trabalho: Luís Miguel Simões de Almeida

2014214377

Renato Paulo Dinis Gomes

2013140694

Índice

Introdução.....	3
Problemas.....	4
1.	4
2.	4
3.	4
3.1).....	4
3.2).....	4
4.	4
4.1).....	4
4.2).....	4
4.3).....	4
5.	5
6.	5
Conclusão.....	6
Bibliografia.....	6

Introdução

Este trabalho pretende analisar e comparar a transmissão de dados usando os protocolos UDP e TCP, recorrendo à plataforma de simulação NS2.

Em síntese, o trabalho depreende-se por o “Servidor 1” enviar ao “Recetor 1” um bloco de dados de 2MB, que começa a ser transmitido no instante 0.5 segundos. Ao mesmo tempo, entre o “Servidor 2” e os recetores 1 e 2, existe tráfego que corresponde a uma stream de dados que está a ser enviada por UDP. Dependendo do cenário considerado a stream de dados está ativa ou desligada.

O **User Datagram Protocol (UDP)** é um protocolo simples da camada de transporte, é um protocolo que visa a velocidade, porém não protege a integridade do que está a ser enviado.

O **Transmission Control Protocol (TCP)** é também é um protocolo simples da camada de transporte, mas que visa a segurança do que é transportado. . Para se certificar que não existe perda de informação este protocolo utiliza uma janela n, que define os pacotes que serão enviados. Depois de enviar os primeiros n pacotes só será enviada nova informação quando o emissor receber o ACK (acknowledge), a informação que os dados enviados chegaram ao destino.

Neste projeto iremos analisar estes dois protocolos (UDP e TCP) ao nível da sua eficiência em termos de duração da transmissão total de dados e ao nível de segurança no envio destes, sendo que a segurança é retratada pelo número de pacotes perdidos.

Problemas:

1. Projeto na sua íntegra presente no ficheiro <project.tcl>.
2. Os dados por omissão usados no NS2 são guardados no ficheiro “./ns-2.35/tcl/lib/ns-default.tcl”.

Tamanho por omissão das filas nos nós	50
Tamanho por omissão dos pacotes TCP	1000
Tamanho por omissão dos pacotes UDP	1000
Tamanho por omissão da janela do TCP	20

3. “Cenário 1”

3.1)

TCP			UDP	
Tempo min	Janela min	Nº pacotes perdidos	Tempo min	Nº pacotes perdidos
0.777538	224	0	0.368368	372

3.2)

TCP			UDP	
Tempo min	Janela min	Nº pacotes perdidos	Tempo min	Nº pacotes perdidos
0.817780	224	0	0.368401	424

4. “Cenário 2”

4.1)

TCP		UDP	
Tempo	Nº pacotes perdidos	Tempo	Nº pacotes perdidos
5.035942	0	0.195040	455

4.2)

TCP		UDP	
Tempo	Nº pacotes perdidos	Tempo	Nº pacotes perdidos
5.076178	0	0.366787	507

4.3)

TCP			UDP		
Tempo min	Janela min	Nº pacotes enviados/recebidos	Tempo min	Nº pacotes perdidos	Velocidade S1-R4
0.913621	164	2099/2099	0.476660	0	37

5. O protocolo UDP é mais rápido na transmissão, porém tem um problema, não possui proteções para as perdas de informação. O protocolo TCP é mais seguro pois realiza um controlo de erros, este controlo que afeta na velocidade de processamento.

De modo geral o melhor protocolo é o do TCP para transmitir um ficheiro através da rede, porque permite que não haja perda de dados, conferindo segurança sobre aquilo que enviamos de forma a que chegue na íntegra ao seu destinatário.

UDP será mais útil em casos onde é feito o envio de dados em tempo real, como por exemplo, em streaming de áudio, vídeo (Skype), etc.

6. O que se sucede é que temos perdas de pacotes aquando do envio. A solução possível é diminuir o tempo de transmissão.

Conclusão

Conclusão retirada nas respostas às questões (Problemas) 5. e 6. .

Como executar:

Na linha de comandos:

- 3.1) (UDP) ns project.tcl 1 udp no 0 1 50
 (TCP) ns project.tcl 1 tcp no 224 1 50
- 3.2) (UDP) ns project.tcl 1 udp yes 0 1 50
 (TCP) ns project.tcl 1 tcp yes 224 1 50
- 4.1) (UDP) ns project.tcl 2 udp no 0 1 50
 (TCP) ns project.tcl 2 tcp no 20 5 50
- 4.2) (UDP) ns project.tcl 2 udp yes 0 1 50
 (TCP) ns project.tcl 2 tcp yes 20 5 50
- 4.3) (UDP) ns project.tcl 2 udp no 0 1 37
 (TCP) ns project.tcl 2 tcp no 164 1 50

Trace-Analyzer

(UDP)-> awk -f trace_analyzer.awk type=cbr src=0 dest=5 flow=1 out.tr

(TCP)-> awk -f trace_analyzer.awk type=tcp src=0 dest=5 flow=1 out.tr

Bibliografia

IRC2015-projeto1_v1.pdf